Linear systems

Numerical analysis - University of Luxembourg

Exercises

Direct solvers

Exercise 1. *Triangular linear systems*
Implement two routines that perform respectively forward and backward substitution. Test your function on two small triangular matrices of your choice.

Exercise 2. *LU decomposition*
Use `scipy.linalg.lu` to compute the L and U factors, and then use your functions from Exercise 1 to solve the linear system thanks to forward and backward substitution Test your code with the matrix

$$A = \begin{pmatrix} 6 & 3 & 4 & 8 \\ 3 & 6 & 5 & 1 \\ 4 & 5 & 10 & 7 \\ 8 & 1 & 7 & 25 \end{pmatrix}$$

and right hand side

$$b = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

Do it again but implement your own LU decomposition (without pivoting) thanks to the pseudo-code seen in class.

Exercise 3. *Pivoting strategy*
Run your linear system solver from Ex. 2 to solve the linear system $Ax = b$, with

$$A = \begin{pmatrix} 10^{-15} & 1 \\ 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Implement a pivoting strategy to handle the issue.

Exercise 4. *QR factorization*
The QR factorization or QR decomposition allows to split the matrix A of size $m \times n$ as a product of unitary $m \times m$ matrix Q with an upper triangular $m \times n$ matrix R, such as

$$A = QR, \quad QQ^T = I, \quad R = \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix},$$

where $\hat{R}$ is of size $n \times n$.

1. Thanks to a backward substitution, write how to solve a square linear system $Ax = b$ in that case.

We will now implement the decomposition thanks to the following steps

1. From the columns $A = [\mathbf{a}_1 \cdots \mathbf{a}_n]$ of the matrix A, write down the Gram-Schmidt procedure and create the orthogonal basis
$$\mathbf{u}_k = \mathbf{a}_k - \sum_{j=1}^{k-1} \text{proj}_{\mathbf{u}_j} \mathbf{a}_k.$$
2. The matrix Q is given by $Q = [\mathbf{e}_1 \cdots \mathbf{e}_n]$, where $\mathbf{e}_k = \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}$.
3. From Q, find how to easily get the matrix R

Finally, recall what are the normal equations (least square problem), and explain how to solve it thanks to QR decomposition. Would your use a QR decomposition rather than a LU decomposition in this case ? Justify your answer
As an application, solve a least square problem and verify your implementation with the built-in `scipy.linalg.qr`.

Iterative solvers

Exercise 5. *A small test*
Consider the linear system

$$\begin{aligned} 7x_1 + 3x_2 + x_3 &= 3 \\ -3x_1 + 10x_2 + 2x_3 &= 4 \\ x_1 + 7x_2 - 15x_3 &= 2 \end{aligned}$$

Questions:

1. write down the k-th Jacobi iteration in its pointwise and matrix form. Specify the matrix splitting of the method. Implement the Jacobi solver and test it,
2. do the same exercise for the Gauss-Seidel method. Comment and explain your results. Which method is fastest and why ?
3. Implement relaxation techniques and experiment with it.